Identifying a Preliminary Circuit for Predicting Gendered Pronouns in GPT-2 Small[1]

Chris Mathwin[2]
Independent Researcher

Guillaume Corlouer[2]
Independent Researcher

Esben Kran[3]
Apart Research

Fazl Barez[3]
Apart Research

Neel Nanda[3]
Independent Researcher

Abstract

We identify the broad structure of a circuit that is associated with correctly predicting a gendered pronoun given the subject of a rhetorical question. Progress towards identifying this circuit is achieved through a variety of existing tools, namely Conmy's Automatic Circuit Discovery and Nanda's Exploratory Analysis tools.

We present this report, not only as a preliminary understanding of the broad structure of a gendered pronoun circuit, but also as (perhaps) a structured, re-implementable procedure (or maybe just naive inspiration) for identifying circuits for other tasks in large transformer language models.

Further work is warranted in refining the proposed circuit and better understanding the associated human-interpretable algorithm.

Keywords: Mechanistic interpretability, ML safety, Circuits, GPT-2 Small

1. Introduction

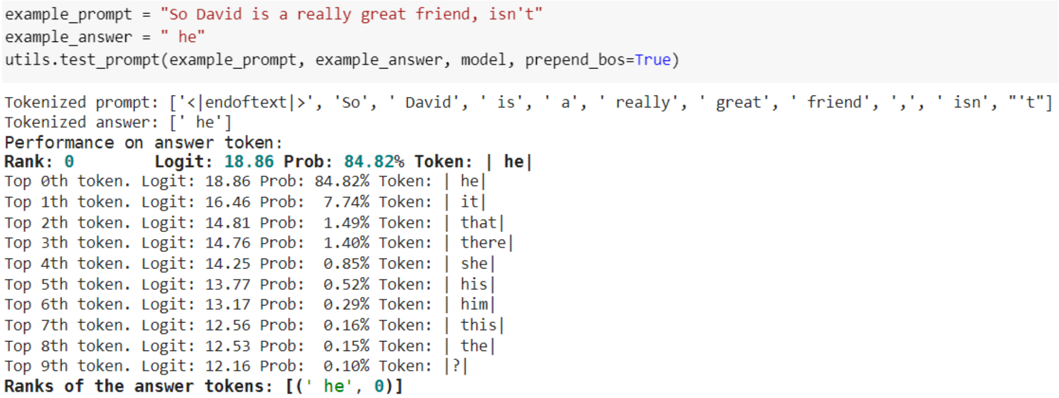
This report presents our high-level understanding of the gendered pronoun circuit implemented by GPT2-Small, the task of predicting a gendered pronoun is best illustrated via example.

> *"So David is a really great friend, isn't" → " he"*
> *"So Mark is such a good cook, isn't" → " he"*
> *"So Mary is a very good athlete, isn't" → " she"*
> *"So Sarah is a really nice person, isn't" → " she"*

[1] Research conducted at the Apart Research Alignment Jam #4 (Mechanistic Interpretability), 2023 (see https://itch.io/jam/mechint)
[2] Equal Contribution Author
[3] Principal Investigator

"So Lisa is such a funny person, isn't" → "she"

The task is thus characterized by predicting the correct 's/he' pronoun given the subject of a rhetorical question of the form "So {name} is a …, isn't {she/he}". The five examples above represent the dataset (with 20 varying male and female names) on which this task is evaluated (100 total prompts). An example of GPT-2 Small performing this task is presented in Figure 1.



```
example_prompt = "So David is a really great friend, isn't"
example_answer = " he"
utils.test_prompt(example_prompt, example_answer, model, prepend_bos=True)
```

```
Tokenized prompt: ['<|endoftext|>', 'So', ' David', ' is', ' a', ' really', ' great', ' friend', ',', ' isn', "'t"]
Tokenized answer: [' he']
Performance on answer token:
Rank: 0        Logit: 18.86 Prob: 84.82% Token: | he|
Top 0th token. Logit: 18.86 Prob: 84.82% Token: | he|
Top 1th token. Logit: 16.46 Prob:  7.74% Token: | it|
Top 2th token. Logit: 14.81 Prob:  1.49% Token: | that|
Top 3th token. Logit: 14.76 Prob:  1.40% Token: | there|
Top 4th token. Logit: 14.25 Prob:  0.85% Token: | she|
Top 5th token. Logit: 13.77 Prob:  0.52% Token: | his|
Top 6th token. Logit: 13.17 Prob:  0.29% Token: | him|
Top 7th token. Logit: 12.56 Prob:  0.16% Token: | this|
Top 8th token. Logit: 12.53 Prob:  0.15% Token: | the|
Top 9th token. Logit: 12.16 Prob:  0.10% Token: |?|
Ranks of the answer tokens: [(' he', 0)]
```

Figure 1 - GPT-2 Small successfully Performing the Gendered Pronoun Task

As can be seen in Figure 1, GPT-2 Small is much more likely to assign ' he' to the next token (85% probability) compared to any other token, and in particular, in comparison to ' she' (0.85% probability).

To more accurately assess the performance of GPT-2 Small on this task, we assess performance by taking the difference between the logits for the correct response (in the above case ' he') and the incorrect response (in the above case ' she'). In this instance, this difference is 18.86 - 14.25 = 4.61. A logit difference of 4.61 can be understood as the model is 100x more likely to predict ' he' than ' she' in this case (via $e^{4.61} = 100$). This behavior is tested more robustly in *Section 2.1*.

Given GPT-2 Small's ability to do this task exceptionally well, we endeavor to identify the circuit that is implementing this behavior. If one views a model as a computational graph with nodes representing blocks (ie. embedding, attention, MLP, unembedding layers) and edges representing the computation that links these nodes, a circuit can be viewed as a computational sub-graph, that is, a distinct subset of nodes and edges within the computational graph. This concept is explored thoroughly within Olah et al.'s *Zoom In: An Introduction to Circuits* work (Olah et al. 2020) and is applied to transformer models in Anthropic's *Transformer Circuits Thread* (Anthropic 2023) and Wang et al.'s *Interpretability in the Wild* paper (Wang et al. 2022). Proceeding sections will document our process for identifying this gendered pronoun circuit and associated findings.

2. Methods

This section is divided into the approximate chronological steps used to identify the broad structure of the gendered pronoun circuit.

2.1. Verifying GPT-2 Small can perform the task.

As outlined within *Section 1,* the first (or zeroth) step toward identifying the gendered pronoun circuit is verifying that the model can indeed do the requisite behavior. We operationalise this behavior as:

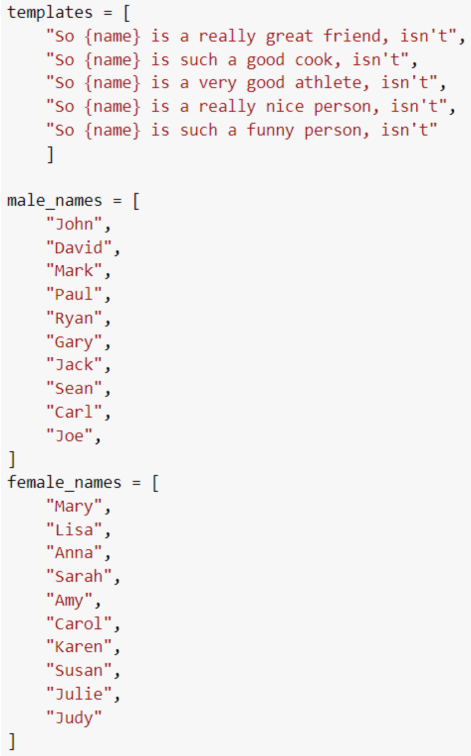
Task: Correctly predicting the gendered pronoun given a subject in rhetorical question-style prompt.

As outlined above, performance on this task can be measured by recording the difference between the logit associated with the correct pronoun (‘s/he’) and the incorrect pronoun (converse ‘s/he’).

Metric: Average difference between Logit[‘correct pronoun’]- Logit[‘incorrect pronoun’]

In section 1, we outlined an example of this metric in use over a single prompt from the dataset (Figure 2).

Original Dataset:



```
templates = [
    "So {name} is a really great friend, isn't",
    "So {name} is such a good cook, isn't",
    "So {name} is a very good athlete, isn't",
    "So {name} is a really nice person, isn't",
    "So {name} is such a funny person, isn't"
    ]

male_names = [
    "John",
    "David",
    "Mark",
    "Paul",
    "Ryan",
    "Gary",
    "Jack",
    "Sean",
    "Carl",
    "Joe",
]
female_names = [
    "Mary",
    "Lisa",
    "Anna",
    "Sarah",
    "Amy",
    "Carol",
    "Karen",
    "Susan",
    "Julie",
    "Judy"
]
```

Figure 2 - Dataset created by replacing {name} with male_name and female_name (creating 100 total prompts, half of which each have a male or female subject).

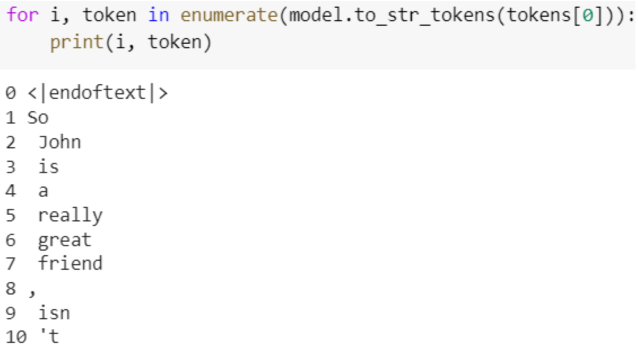
Using the dataset outlined in Figure 2, we can more robustly test model performance on this task by performing this calculation over the 100 total prompts and averaging the logit difference found between correct and incorrect responses, we find that GPT-2 Small has an average logit difference of 4.73 for this task. This means that GPT-2 Small is, on average, 113x more likely to choose the correct gendered pronoun over the incorrect gendered pronoun for the 100 prompts, GPT-2 Small is reliable at this task.

2.2. Using Conmy's Automatic Circuit Discovery Tool

We use Conmy's Automatic Circuit Discovery tool to identify the general sub-graph for this behavior within GPT2-Small, resources for this tool can be found in way of a demonstration notebook (Conmy 2022) and descriptive blog post (Conmy 2022). The tool uses direct path patching, a powerful variant of activation patching, that allows for identification of model components that have the largest influence of logit differences on a given task. To carry out the Automatic Circuit Discovery (Conmy 2022), a number of steps are required.

1. **Create a dataset (original dataset) to succinctly capture the behavior.**
   In the gendered pronoun case, this dataset is described in Section 2.1 and represents the 100 prompts.

2. **Label important positions within the structure of the sequence.**
   For this reason, it is important that each of the tokens within the original dataset have consistent positioning of important features. We hypothesize that the 'name', 'is', 'person' (friend/cook/athlete/person/person), 'isn' and ''t' are all important positions for the circuit to use in eliciting the desired behavior. We record the position of these tokens (Figure 3) and construct an ordered dictionary that the Automatic Circuit Discovery tool will use to distinguish which token positions to evaluate each head's (or other components') effect on logit difference.



```
for i, token in enumerate(model.to_str_tokens(tokens[0])):
    print(i, token)
```

```
0 <|endoftext|>
1 So
2  John
3  is
4  a
5  really
6  great
7  friend
8 ,
9  isn
10 't
```

Figure 3 - Token Positions for Important Features in the Original Dataset

Using the position labels corresponding with the above noted features produces labels of 2, 3, 7, 9, 10. These form key:value pairs in the ordered dictionary, wherein the key corresponds to the feature label and the value corresponds to a row-vector of position labels with length equal to the number of prompts (ie. 'name' : [2, 2, …, 2]).

3. **Make a baseline dataset.**
   This represents a dataset which has the same structure as the original dataset, but of which, the model would be unable to infer the subject's gender (and thus unable to correctly choose between ' she' and ' he'). This baseline dataset is used within the direct path patching algorithm as patch in place of the original dataset to corrupt specific activations. In our specific case, the prompt that we chose was,

Baseline Dataset Prompt:
"That person is a really great friend, isn't"

In constructing the baseline dataset, we simply repeat this dataset such that it matches the batch size (100 examples) of our original dataset.

4. **Define a metric to measure the model's performance on the task.**
   This metric has been discussed in previous sections, our metric is the difference between the logit associated with the correct response ('she' or 'he') and the logit associated with the incorrect response (the converse 'she' or 'he') for a given prompt. Further details of this metric can be found in *Section 2.1*.

5. **Choosing a threshold value.**
   The threshold value corresponds to the minimum value from the associated attention and MLP results to consider as part of the computational sub-graph. In our case, this required some experimentation, we found that a threshold value of 0.015 was appropriate in extracting the gendered pronoun circuit but present a number of thresholds (0.02, 0.015, 0.01 and 0.005) within *Section 3* for comparison. These different thresholds represented a search for a reasonable tradeoff between finding a minimal circuit and a circuit that possesses good performance.

6. **Construct the Circuit object and run the .eval method.**
   After defining each dataset, the position labels and the performance metric, we can simply construct a Circuit object using Conmy's Automatic Circuit Discovery tool (Conmy 2022).

   Once constructed, we can call **.eval** to derive the computational sub-graph via direct path patching (Conmy 2022) .

   Setting show_graphics = True will produce plots of the path patching graphics while the circuit object is being constructed; these will be useful for future analysis in understanding what aspects of the computational sub-graph are most significant.

   For the 0.015 threshold, this computational sub-graph is presented in Figure 4.

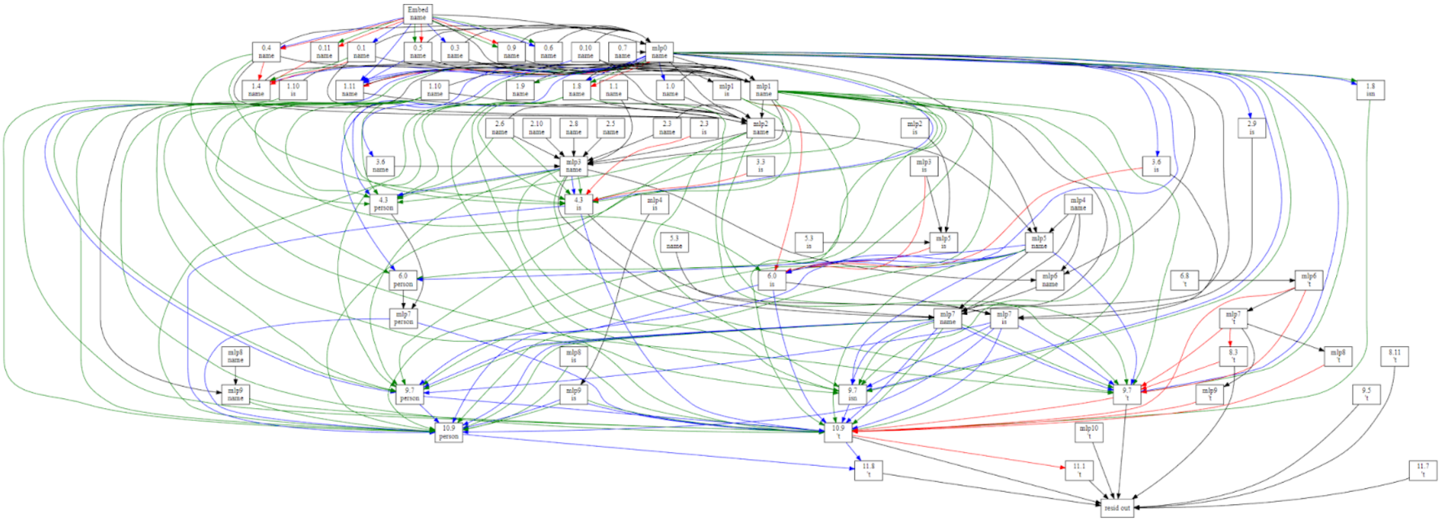


Figure 4 - Circuit derived with Conmy's Automatic Circuit Discovery Tool

A complete description of interpreting that circuit extracted from the Automatic Circuit Discovery tool can be found in Conmy 2022, but in essence, each node represents a computational unit (specific head, MLP, etc.) at a specific token position and nodes represent the computation performed linking these nodes. Specifically, green edges represent key composition, red edges represent query composition and blue edges represent value composition.

7. **Evaluate the circuit's performance with respect to the entire model's performance.**

With the circuit derived (Figure 4), the isolated circuit can be evaluated on the original dataset, the 100 prompts, and its performance compared to the performance of the complete model. This can be achieved by calling the **evaluate_circuit** function on the Circuit object produced in Step 6.

It was found that circuit produced in Step 6 (and presented in Figure 4) produces an average logit difference across the 100 prompts of 3.08, indicating that on average, this isolated sub-graph is 21x more likely to choose the correct pronoun ('s/he') associated with a subject in a rhetoric question than the incorrect pronoun.

In comparison, this circuit represents 65% of the model's full performance while only using approximately 5% of the model's (head, token) pairs. It can thus be deduced that a significant component of the model's performance on this gendered pronoun task resides in this subset of the computational graph.

2.3. Identifying significant heads to investigate via Direct Logit Attribution and Path Patching plots

To improve our ability to identify a human-interpretable algorithm from the circuit identified in *Section 2.2*, we endeavor to identify the (head, token) pairs within the circuit that most significantly affect logit difference. This is achieved by first applying the Direct

Logit Attribution tool from Nanda's Exploratory Analysis tooling (Nanda 2023) and then using the path patching graphics derived during Conmy's Automatic Circuit Discovery in *Section 2.2*

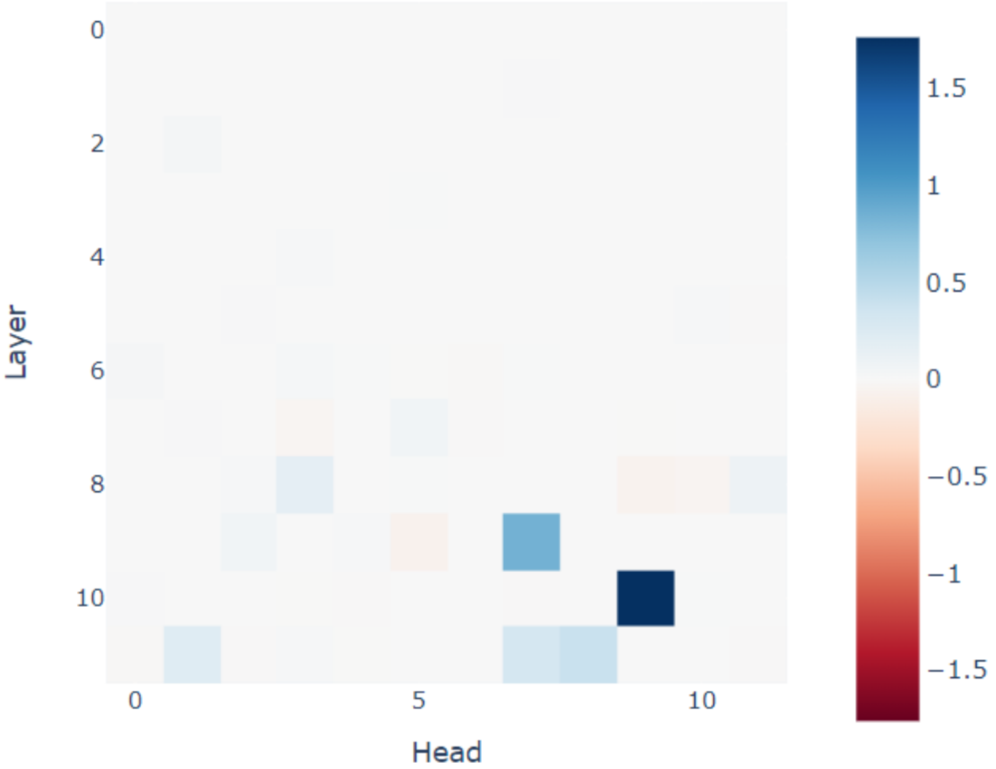


Figure 5 - Direct Logit Attribution

We can see that 2 heads possess significant contributions towards the logit difference for the gendered pronoun task, these heads are:

- Layer 10, Head 9
- Layer 9, Head 7

We find that each of these heads play a direct role in 'resid_out' (and thus output logits) via the "'t" token. For simplicity, we have also decided to ignore the effect of MLP blocks at this stage. This leaves L10H9 and L9H7 and all of their attention-block dependencies as our first line of inquiry.

Further, by inspecting the direct path patching graphics associated with Conmy's Automatic Circuit Discovery Circuit (from Figure 4), we can confirm that the (head, token) pair that has the largest effect on logit difference is (10.9, 't) and (9.7, 't), this is presented in Figure 6.

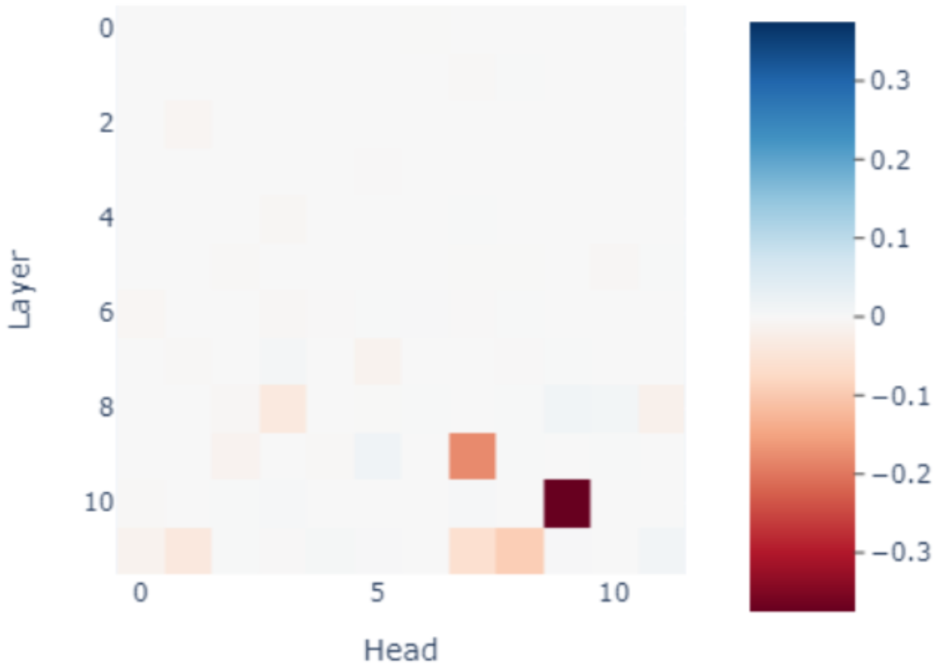


Figure 6 - Path Patching to from each (layer, head) at position ‘t to ‘resid out’

We can now begin recursively path-patching back through the computational sub-graph, identifying the heads that have the largest effect on proceeding heads. For example, having identified (Layer 10, Head 9, token ‘t) as having a particularly important effect on logit difference, we can inspect the path-patching graphics that feed into this (head, token) pair.

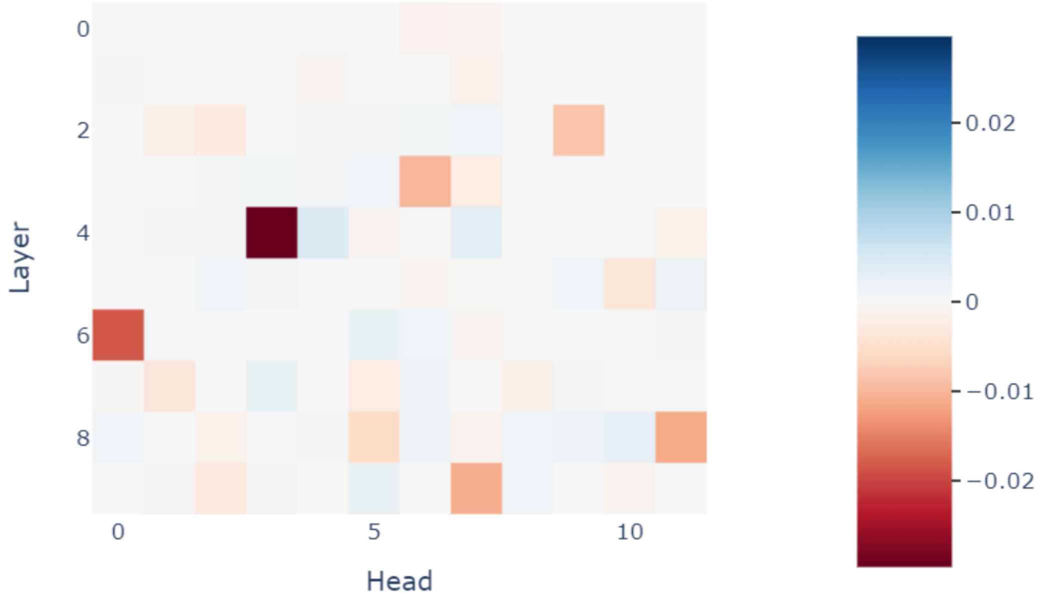


Figure 7 - Path Patching to from each (layer, head) at position ‘is’ to (10.9, ‘t)’s Value Vector

We find that a number of heads significantly affect the Value vector of the (10.9, ‘t) pair, in particular the (6.0, ‘is’) and (4.3, ‘is’) pairs (Figure 6).

We can continue to recursively path-patch back through the computational sub-graph, now focusing on the (4.3, ‘is’) pair and deducing which (head, token) pairs have the largest effect on this pair.

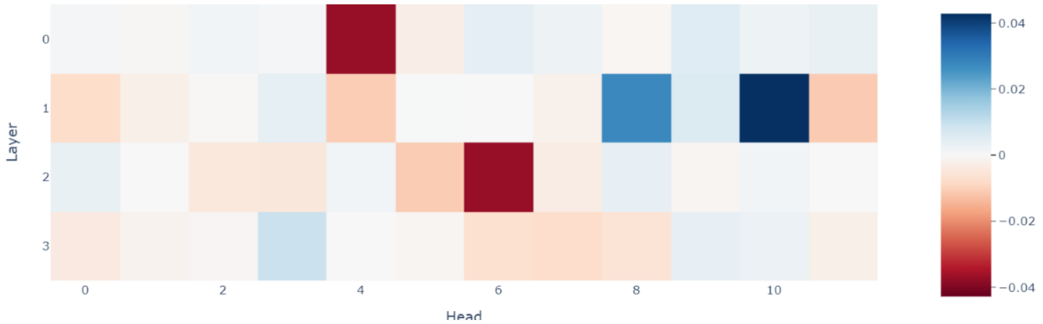


Figure 8 - Path Patching to from each (layer, head) at position ‘name’ to (4.3, ‘is’)’s Key Vector

We find that a number of heads significantly affect the Key vector of the (4.3, ‘is’) pair, namely the (0.4, ‘name’) and (2.6, ‘name’) pairs.

We continue this same exercise of recursively investigating the heads that have the largest influence via these path-patching plots. We find that there appears to exist 3 groups of significant (head, token pairs).

The ‘name’ group:
These include (but are not limited to):

- (0.4, ‘name’),
- (1.4, ‘name’) and,
- (2.6 ‘name’)

This group influences the ‘is’ group via the ‘is’ group’s key vectors.

The ‘is’ group:
These include (but are not limited to):

- (4.3, ‘is’) and,
- (6.0, ‘is’)

This group influences the ‘’t’ group via the ‘’t’ group’s value vectors.

The ‘’t’ group:

These include (but are not limited to):

- (10.9, ‘t)

This group directly influences the residual stream output.

2.4. Attention patterns in significant heads

Attention visualization allows looking at how an attention's head moves information between tokens. For example, during direct logit's attribution, we can look at the attention pattern from the final token.

Attention visualization complements the circuit detection procedure, knowing significant heads in a circuit, we can infer the information flow between tokens for each head and explore whether patterns of information corroborate the information flow in the computational subgraph.

We refer to our Colab notebook to inspect attention patterns analysis on significant heads, and after direct logit attribution (our notebook is an adaptation of the attention analysis section in the demo notebook for exploratory analysis of circuits). We report the result of attention visualization in the results section 3.3.

3. Results

3.1. Circuit identification

An important consideration when running Conmy’s Automatic Circuit Discovery tool is the selection of a threshold, as discussed in *Section 2.2.* Through the process of experimenting with different thresholds and recording the resulting circuit performance and size (relative to the model), we found a number of interesting results.

We initially started with an acceptance threshold value of 2%, we found that the circuit derived performed poorly with a relative performance of 3% compared to the actual model and with 26 (component, token) pairs. This circuit is presented in Figure 9.

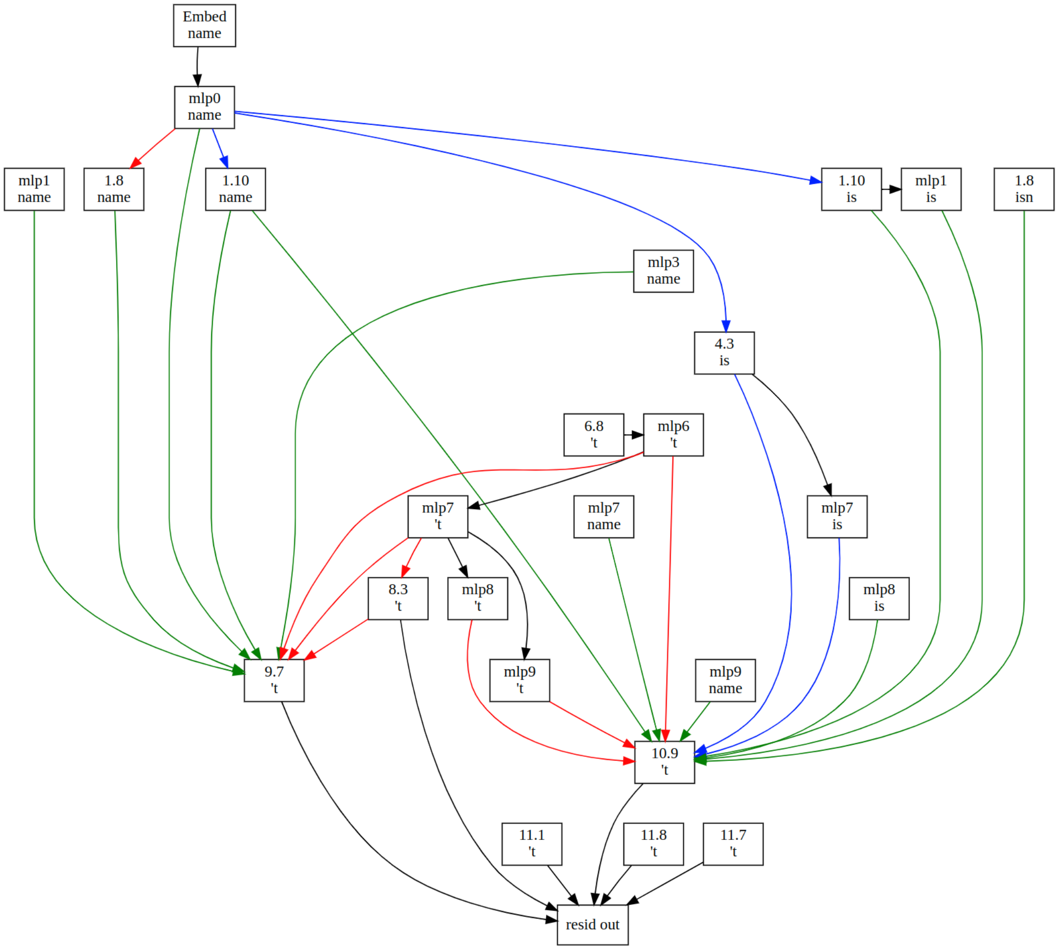


Figure 9 - Circuit derived with Conmy's Automatic Circuit Discovery Tool using a 2% threshold

We then explore the 1.5% threshold that is presented in *Section 2.2*, this circuit performs substantially better and achieves a logit difference performance of 65% relative to the model's performance with 72 (component, token) pairs.

We also explore a 1% threshold circuit and find that although it is larger than the 1.5% circuit with 159 (component, token) pairs, it interestingly performs better than the model itself on this task, achieving 122% of model performance. Over the 100 prompts, this circuit achieved an average logit difference of 5.77, indicating that it is 321x more likely to select the correct pronoun relative to the incorrect one (for reference, the model was 133x more likely to choose the correct pronoun relative to the incorrect pronoun). We expect that this is suggestive of the model perhaps having some components that are inhibitory of performance, perhaps analogous to the 'Negative Name Movers' found by Wang et al. (2022) in their IOI circuit. Further investigation into the role of the suspected presence and identification of these inhibitory heads in the gendered pronoun circuit would be an interesting direction.

3.2. Most significant heads

As presented within *Section 2.3* it is possible to speculate from Conmy's Automatic Circuit Tooling, Direct Logit Attribution and Path Patching plots the heads that perform the most significant roles within the gendered pronoun circuit. As presented within *Section 2.3*, these can be summarized tentatively within Figure 10.

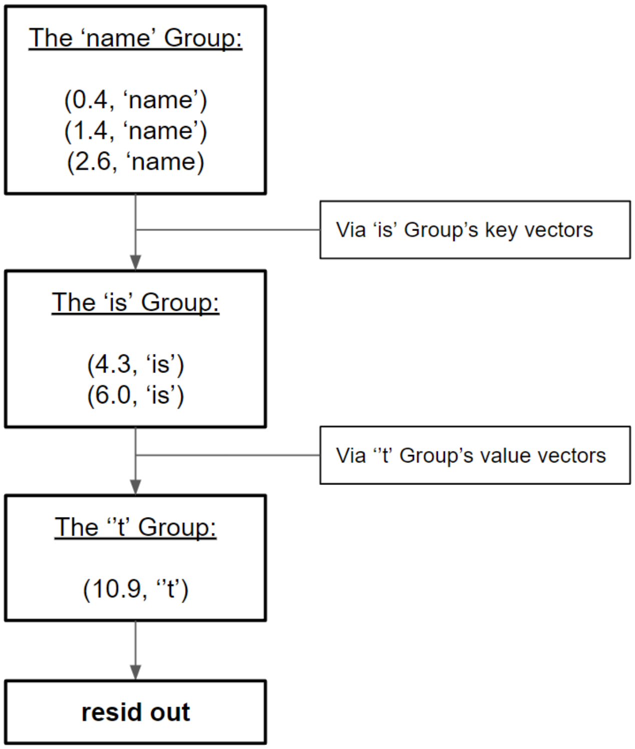


Figure 10 - A tentative description of the most significant heads and their role within the gendered pronoun circuit

3.3. Salient attention patterns

We report the strongest attention patterns for the most significant heads in Early (0 to 3) Middle (4 to 7) and Late layers (8 to 12). We refer to the attention patterns in our Colab notebook (adapted from the exploratory analysis demo).

To improve understanding, we have adopted the following terminology for describing positions within the attention matrices:

- 'name' → S
- 'is' → V1
- 'isn't' → V2
- ''t' → END

Top early heads: L0H4, L1H8, L1H10, L1H4, L2H6, L3H6.

- L0H4: Self-attending to S
- L1H8: No effect
- L1H10: Self-attending to S
- L1H4: No effect

- L3H6: Token V1 attending to S

Top early heads either have no interesting attention pattern or are self attending to S token. The “deepest” early layer L3H6 has information flow from V1 to S, which is also a pattern that we see in the middle heads.

Top middle heads: L4H3, L6H0.

- L4H3: V1 attending to S (more generally all tokens attend to S but the magnitude of information flow was less strong than V1 to S)
- L6H0: V1 attending to S

  Top middle heads show information flow from V1 to S.

Top late heads: L9H7, L10H9, L11H8

- L9H7: END attends to S
- L10H9: END attends to V2, END attends to V1; and V2 attends to S
  - Although strongest seems to be END to V2
- L11H8: Self-attending to V2

Strongest pattern from END token to S and from END to V2. Also from END to V1 and V2 to S but difficult to say whether these effects are “significant”.

4. Discussion and Conclusion

Summary

Within this report, we aim to present both a step-by-step (but potentially naive) method for circuit identification and a first-pass at identifying a gendered pronoun circuit.

This step-by-step (but again, potentially naive) method consists of using various existing tools, namely Nanda’s TransformerLens (Nanda 2023), Conmy’s Automatic Circuit Discovery and various tools from Nanda’s Exploratory Analysis notebook, namely Attention Patterns and Direct Logit Attribution.

The gendered pronoun circuit is identified within *Section 2.2* and consists of a circuit that retains 65% the performance of the original model while only requiring approximately 5% of the (component, token) pairs. We then endeavor to make progress toward a human-interpretable algorithm by identifying which aspects of the computational sub-graph are most significant. This allows us to speculate on the fundamental information flow that the gendered pronoun circuit may be implementing, this is presented again in Figure 11.

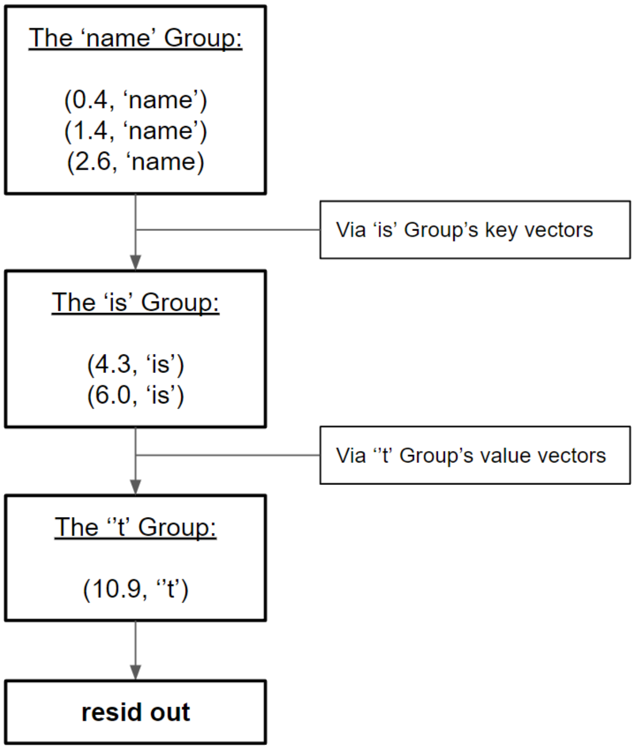


Figure 11 - A minimal circuit for information flow based on most significant heads

We speculate that, fundamentally, the gendered pronoun circuit is moving information about the gendered pronoun task from the 'name group', a collection of early layer heads on the 'name' position, to the 'is' group via the 'is' group's key vectors. The 'is' group, a collection of middle layer heads on the 'is' position, is then moving information to the ''t' group via the ''t' group's value vectors. The ''t' group, a collection of late layer heads, then directly affects the model's logit scores.

Consistency between computational graph and attention patterns

The attention patterns that we found in the most important Early and Middle heads were consistent with the computational graph. The strongest effects in the most important Early heads showed that they self-attended to S and the computational graph shows information flowing between the S token within early Heads or between the embedding and Early heads.[4] The strongest attention patterns from Middle heads showed that L4H3 and L6H0 heads attend heavily from the 'is' token to the 'name token. This suggests that heads from Middle attention layer 'is' are attending to information from 'name' group heads in earlier networks.

Limitations

While we have identified a circuit of the model that performs relatively well at the task of predicted gendered pronouns in rhetorical questions, we have not been able to reverse engineer the computations that the computational graph of most important heads are

[4] Hand-wavy speculation: Perhaps the "self-attention" is related to the embedding in some way…..

implementing. The minimal circuit presented in Figure 11 suggests that S, V1 and END play a particularly important role in the prediction of gendered pronouns–which makes sense intuitively–but the function of each head and layer remains to be identified.

The current procedure also lacks a more "objective" criteria to choose the appropriate threshold for circuit identification. For example our results suggested that taking a threshold of 1.5% yielded the best ratio between a small circuit size and task-performance but it might be that other criteria are important.

Finding the most important heads rely on measuring the influence between heads via path patching. While path patching allows to give a magnitude of influence it would be interesting to find a more "objective" threshold allowing to systematically detect when the influence of one heads over another is significant[5]. Finding "significant" attention patterns also faces the same issue. More generally this points to a limitation of current interpretability techniques: the lack of a principled criteria to include neurons/heads in a circuit and seems to be mostly based on magnitudes in specific contexts.

Importantly, for simplicity, we ignored the role of MLPs' layers and neurons in the computational graph. In contrast to the IOI tasks which inspired our project, MLPs are involved in many steps of the computational graph that we identified which suggest that non linear operations are important to understand the algorithm implemented by a circuit involved in the prediction of gendered pronouns.

More generally, identifying the most important heads identifies the influence between pairs of heads. It would also be interesting to look at the influence of groups of heads. It may be misleading to look at information flow only between pairs of heads and instead try to consider groups of heads as computational units between which information is flowing.

Further Research

There a number of interesting directions for future research associated with this work, these include:

- Further work is warranted in refining the circuit proposed in this report and reverse engineering the associated algorithm.
- The role of MLPs within this circuit have not been considered, further work is warranted in understanding the function of MLPs within the circuit.
- A circuit was identified that performed better on the pronoun task than the original model (the 1% threshold circuit), this may indicate that aspects of the model are performing inhibitory roles, perhaps analogous to Negative Name Movers in the IOI circuit identified in the Interpretability in the Wild paper (Wang et al. 2022). Further work is warranted in understanding why the 1% threshold circuit possessed performance greater than that of the original model

[5] Something similar to hypothesis testing or calculating bayes factors

and the presence and potentially the function of heads inhibiting performance on the pronoun task.

- Further work is warranted in understanding specific roles of both the QK and OV circuits within the circuit produced within this report.

- This work employed subjective assessments of which (head, token) pair and attention patterns are deemed to have significant effect on model performance. An objective criteria to selecting which (head, token) pairs and attention patterns are deemed significant is warranted.

5. References

Anthropic. (2023). Transformer Circuits Thread. https://transformer-circuits.pub/

Comney, A. (2022). Automatic Circuit Discovery Notebook. https://colab.research.google.com/github/ArthurConmy/Easy-Transformer/blob/main/AutomaticCircuitDiscovery.ipynb

Comney, A. (2022). Automatic Circuit Discovery.. https://arthurconmy.github.io/automatic_circuit_discovery/

Nanda, N. (2023). TransformerLens. https://github.com/neelnanda-io/TransformerLens

Nanda, N. (2023). Exploratory Analysis Demo Notebook. https://colab.research.google.com/github/neelnanda-io/Easy-Transformer/blob/main/Exploratory_Analysis_Demo.ipynb#scrollTo=2EmZ15ejPTVo

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom In: An Introduction to Circuits. *Distill*, *5*(3), 10.23915/distill.00024.001. https://doi.org/10.23915/distill.00024.001

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., & Steinhardt, J. (2022). Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. https://arxiv.org/pdf/2211.00593.pdf